

RAZVOJ OBRAZOVNE IGRE U JAVAFX TEHNOLOGIJI EDUCATIONAL GAME DEVELOPMENT IN JAVAFX TECHNOLOGY

Miroslav S. Slavković, Đorđe M. Đurđević, Igor I. Tartalja
Univerzitet u Beogradu - Elektrotehnički fakultet

REZIME: Proces učenja koji uključuje sticanje znanja i veština se značajno unapređuje uvođenjem modernih tehnologija i zabavnih sadržaja, a obrazovna računarska igra ostvaruje njihov sinergistički efekat. Jedna od prvih modernih ideja o upotrebi računarskih igara u obrazovanju pojavila se 1974. godine. Danas se nekoliko stotina institucija bavi istraživanjem i razvojem obrazovnih igara u celom svetu. Na Elektrotehničkom fakultetu u Beogradu razvijen je veći broj prototipova obrazovnih igara, a 2010. godine razvijena je prva verzija softverskog paketa Olimpijada znanja, koji se sastojao od dve aplikacije. Aplikacija Editor je namenjena instruktorima za kreiranje pojedinih igara, svojevrsnih kvizova znanja u proizvoljnoj oblasti, kombinovanih sa konceptima igre na sreću. Aplikacija Takmičenje namenjena je igračima koji kroz igru stižu i proveravaju svoje znanje. Tema ovog rada je razvoj nove verzije aplikacije Takmičenje sa elementima veb-orijentisane aplikacije, kao i njena konverzija iz Java2D u moderniju tehnologiju JavaFX. Nova verzija aplikacije Takmičenje je urađena u dve varijante: kao konvencionalna desktop aplikacija i kao aplikacija koja se pokreće iz čitača veba i koristi resurse sa servera na Internetu. Varijante aplikacije se neznatno razlikuju, u svega nekoliko desetina redova izvornog koda.

KLJUČNE REČI: Obrazovanje kroz zabavu, Obrazovna računarska igra, Kviz, Bogata internet aplikacija, JavaFX aplikacija.

ABSTRACT: The process of learning, which includes knowledge and skill acquisition, is considerably improved by introduction of modern technologies and entertaining contents, and an educational computer game achieves their synergistic effect. One of the first modern ideas about use of computer games in education appeared in 1974. Today, several hundreds of institutions worldwide are engaged in the development of educational games. Several prototypes of educational games were previously developed at the University of Belgrade - School of Electrical Engineering, including the first version of software package Game Olympics in 2010, which consists of two applications. The Editor application is intended to be used by instructors to create particular games, which essentially are quizzes in an arbitrary knowledge area, combined with concepts of games of chance. The Contest application is intended to be used by players, who learn and test their knowledge by playing. This paper is focused on the development of the new version of the Contest application, with elements of web-oriented application, as well as its conversion from Java2D to the modern JavaFX technology. The new version comes in two forms: as a conventional desktop application, and as web-browser based application, which uses resources from an Internet server. The difference between both forms is negligible, only about a several dozens of lines of source code.

KEY WORDS: Edutainment, Educational computer game, Quiz, Rich Internet application, JavaFX application

1. UVOD

U procesu učenja, čiji je cilj sticanje znanja i veština, postoji više elemenata koji olakšavaju i ubrzavaju put do tog cilja, uz pozitivan uticaj na kvalitet samog procesa učenja, kao i na kvalitet stečenih znanja i veština. Jedan od elemenata je upotreba savremenih tehnologija, kakve su računarske i druge digitalne tehnologije. Drugi element je uvođenje zabavnih sadržaja. Ta dva elementa mogu se posmatrati odvojeno. Na primer, obogaćena stvarnost (eng. *augmented reality*) je pristup u učenju kod kojeg onaj koji uči interaguje sa gradivom, uveden je u sam proces učenja i aktivno učestvuje [1], što svakako pospešuje proces učenja, ali ne mora da ima nikakve zabavne elemente. S druge strane, zabavni sadržaji ne moraju da budu podržani tehnološkim pomagalima, ali se mogu koristiti u nastavi. Nastavnici ponekad organizuju zabavne događaje, kao što su društvene igre, i kroz njih učenicima ponude nova saznanja. Međutim, uzajamna dopuna ova dva elementa je veoma zanimljiva, jer omogućava efikasno usvajanje znanja uz zadovoljstvo učenika. Kao problem se postavlja kako iskoristiti moderne tehnologije u oblikovanju procesa učenja tako da on bude zabavan za one koji uče. Prirodno rešenje ovog problema, koje treba da proizvede očekivani sinergistički efekat, nalazimo u računarskim obrazovnim igrama. Jednoj takvoj igri i njenom razvoju je posvećen ovaj rad.

Upotreba modernih tehnologija u obrazovanju je sasvim prirodna težnja, jer one značajno ubrzavaju i, dugoročno gledano, pojeftinjuju praktično sve procese u koje se uvode. Ni obrazovanje kroz zabavu nije nova tema. Još je Džon Lok 1693. godine pominjao mogućnost da se deca mogu podučavati uz igru i relaksaciju [2]. Iako se može postaviti neka vrsta granice između zabave namenjene učenju i one koja to nije, ne treba isključiti mogućnost da se nešto može naučiti čak i kroz ovu drugu. Ipak, neki autori smatraju da zabavni sadržaji osmišljeni sa primarnim ciljem da se koriste u obrazovanju dece ne treba da skrivaju svoju pravu namenu, jer decu nije jednostavno prevariti [3]. U [4] se navode tri dobrobiti upotrebe igara u nastavi:

- ponavljanje – igre mogu ponavljati neke obrazovne sadržaje proizvoljan broj puta i tako pomoći u njihovom usvajanju;
- nagrada – poeni osvojeni u igri i dospevanje na rang listu su svojevrsna nagrada za igrače i značajan motivator;
- razmišljanje – igra može zahtevati od igrača da se mentalno uključi u rešavanje raznih problema i na taj način igra postaje intelektualni stimulans.

Većina digitalnih igara koje se koriste u obrazovne svrhe su razvijene na univerzitetima širom sveta. Postoje i komercijalne obrazovne igre, ali su one ipak razvijene sa primarnim

ciljem da donesu profit tako da im je obrazovna komponenta ponekad nije u prvom planu. Na Elektrotehničkom fakultetu u Beogradu 2010. godine razvijena je prva verzija obrazovne digitalne igre *Olimpijada znanja*, u daljem tekstu OZ 1.0. U pitanju je 2D igra za jednog ili više igrača koja može da služi za uvežbavanje i proveru znanja kroz kombinovanje kviza znanja sa elementima igre na sreću. OZ 1.0 je sistem od dve desktop aplikacije, jedne za instruktora (*Editor*), druge za obučavanog (*Takmičenje*), od kojih ni jedna ne koristi veb tehnologije. Razvoj ove igre opisan je u [5, 6].

Veb postaje sve atraktivnija platforma za aplikacije u mnogim sferama. Sve aktuelniji bivaju računarstvo u oblaku (*cloud computing*) i softver kao servis (*software as a service*). Korisnik ne mora čak ni da kupi i instalira neki paket (na primer, *Microsoft Office*), može da ga zakupi i koristi na Vebu (*Office 365*). Takođe, korisnik ne mora da čuva svoje dokumente na svom računaru, može ih čuvati i koristiti preko posebnih servisa (kakav je, na primer, *Dropbox*) i na taj način može da pristupa istim datotekama s bilo kog računara (na primer od kuće ili sa posla), tableta ili mobilnih telefona na kojima je operativni sistem *Android*, *iOS*, *Windows* ili neki drugi. Poslovni model se menja i polako okreće ka Vebu – sve više malih kompanija započinje svoje poslovanje sa nekom novom idejom orijentisanom prema Vebu i odmah dobija ogromno otvoreno tržište za svoj proizvod.

Desktop igre danas imaju veoma konkurentno tržište, čiju ciljnu grupu čine pasionirani igrači. Iako moraju da se instaliraju na računar korisnika, desktop igre sve više koriste Veb na razne načine. Prvo, desktop igre se sve više distribuiraju preko Veba. U veb-prodavnici *Steam* [7] igra se može kupiti istog dana kada se pojavi, igrači ne moraju da čekaju da se dopreme diskovi u prodavnice. Postoje i igre koje se distribuiraju isključivo preko Veba. *Sony* je objavio da će za njihovu novu konzolu *PlayStation 4* sve igre biti dostupne na Vebu. Drugo, kod desktop igre za više igrača, komunikacija između računara može da se odvija preko Interneta. Jedan ilustrativan primer ovoga je *Left 4 Dead*, igra za do 4 igrača koja se prodaje sa 4 licence. Ovako kupljena igra se instalira na 4 računara i 4 igrača je igraju međusobno, povezani Internetom. Treće, mnoge desktop igre obezbeđuju globalne rang liste. Četvrto, postoje igre kao što je *World of Warcraft*, gde avatari obavljaju određene aktivnosti („žive na Vebu“) čak i kad njihovi vlasnici ne igraju igru.

Veb igre, iako u početku tehnološki inferiorne u odnosu na desktop igre, danas se nalaze u velikoj ekspanziji. To je naročito vidljivo na društvenoj mreži *Facebook*. Kompanija *Electronic Arts*, koja je svetski lider u proizvodnji desktop igara, istog dana je otpustila 1500 radnika i kupila kompaniju *PlayFish*, koja se bavi izradom *Facebook* igara, za 300 miliona dolara [8]. Ciljna grupa kod veb igara je povremeni igrač koji nema vremena da preuzima sa Veba ili da ode u radnju i kupi igru, a zatim i instalira igru, koji će je igrati povremeno i koji može da je igra na svakom mestu na kojem postoji Internet i računar ili drugi uređaj sa čitačem veba (eng. *web browser*). I zaista, veb-igre sve ovo nude: ne mora ništa da se preuzima i instalira, samo se pokrene čitač veba, unese URL (adresa na Vebu) i igra

može da počne. Osim toga veb-igra na društvenoj mreži uključuje i socijalnu komponentu tako da igrači igraju sa svojim „prijateljima“ na društvenoj mreži i međusobno se takmiče.

Proizvođač vrhunskih grafičkih čipova NVIDIA istražuje u pravcu „crtanja u oblaku“ (eng. *cloud rendering*), odnosno radi na ideji da se na serveru generišu slike i gotove šalju igrama na računarima povezanim na Internet. Takve slike bi mogle biti značajno većeg kvaliteta nego što bi se dobile u prihvatljivom vremenu na računarima sa slabijom grafikom. Već postoji nekoliko veb-portala koji su implementirali „igranje u oblaku“ (eng. *cloud gaming*) i na kojima se mogu igrati igre čija se grafika iscrtava na serveru, dok se igra izvršava na desktop računaru.

Razvoj OZ okrenute ka Vebu, u svetlu novih trendova, je prirodan korak u razvoju ove igre. OZ 2.0 predstavlja prvi korak tog razvoja. U tom cilju izvršena je konverzija aplikacije *Takmičenje* u moderniju tehnologiju koja je omogućila da se pored desktop varijante dobije i varijanta aplikacije koja koristi resurse sa servera na Vebu. Dok se desktop varijanta za krajnjeg korisnika gotovo ne razlikuje od OZ 1.0, veb varijanta aplikacije se pokreće iz čitača veba. Ovaj rad analizira primenjive tehnologije, obrazlaže projektne odluke i opisuje razvojna iskustva stečena kroz konverziju obrazovne igre OZ 1.0 iz Java2D tehnologije u JavaFX tehnologiju. Iako je predmet rada razvoj nove verzije obrazovne igre, sprovedena analiza, diskusija odluka i stečena iskustva mogu biti od interesa za širu populaciju projekatanta i programera suočenih sa srodnim problemima uporednog razvoja desktop i veb varijanti aplikacija, uz nastojanje da funkcionalne i strukturne razlike između ovih varijanti budu što manje.

U poglavlju 2 se daje osvrt na problem rešavan razvojem OZ 1.0, a zatim se prelazi na problem koji je rešavan kroz razvoj OZ 2.0. Daje se i osvrt na preostale probleme koji treba da se reše u budućim verzijama OZ. U poglavlju 3 je prvo prikazano trenutno stanje obrazovnih igara u svetu, a zatim je urađena uporedna analiza OZ i dve njoj srodne igre. Iako se OZ 2.0 veoma malo razlikuje u koncepciji igre i svim njenim funkcionalnostima u odnosu na OZ 1.0, u poglavlju 4 ukratko su opisane koncepcija i funkcionalnosti OZ. U poglavlju 5 su prikazane trenutno aktuelne tehnologije za veb aplikacije, sa posebnim akcentom na bogate internet aplikacije (eng. *Rich Internet Application*, RIA), kao i na izbor tehnologija za razvoj OZ 2.0. U poglavlju 6 je opisan projekat softvera i dati su implementacioni detalji. U zaključku je napravljena retrospektiva rezultata rada i date neke smernice za dalji razvoj OZ.

2. PROBLEM

Popularno rešenje problema provere znanja kroz igru je kviz. Naravno, suvoparno postavljanje pitanja jednog za drugim igru ne bi činilo zanimljivom. Elementi igre na sreću sa nekim slučajnim događajima unose zanimljivost koja privlači i drži pažnju igrača. Tako se došlo do ideje da se u igri iskoristi *linearna otvorena putanja* kojom se napreduje bacanjem kocke. Primeri linearnih putanja su kod stonih igara *Monopol* i *Ne ljuti se čoveče*. Poenta kod ovih igara je da od ishoda na kocki

zavisi za koliko polja će igrač da se pomeri unapred, a polje na koje igrač na kraju dospe određuje dalji razvoj igre. Kod igre *Monopol* se koristi zatvorena putanja, tj. nema cilja na putanji, već igrači kada pređu celu putanju nastavljaju istom putanjom ponovo, u krug. Kod *Ne ljuti se čoveče*, iako je globalna putanja zatvorena, za pojedinog igrača je ona otvorena, postoje početna i ciljna pozicija i kada figura igrača dospe do cilja ona završava svoj put.

OZ je igra sa kombinacijom linearne otvorene putanje i kviza. Kako igrači napreduju prema cilju bacanjem kocke, događa se da povremeno stanu na polja koja će inicirati postavljanje pitanja igraču, a od njegovog odgovora zavisi dalji tok igre. Detaljna funkcionalna specifikacija OZ je data u [9]. Izgled putanje i pitanja koja se postavljaju nisu predefinisana, već je omogućeno instruktoru da ih kreira i prilagodi ciljnoj populaciji igrača. Paralelno sa razvojem aplikacije za učenika *OZ Takmičenje 1.0*, razvijena je aplikacija *OZ Editor 1.0* koja služi instruktorima da kreiraju putanje sa poljima, skupove pitanja sa tačnim i netačnim odgovorima i da parametrizuju pravila igre. Putanju, pitanja i pravila koja čine jednu pojedinačnu igru nazivamo *tablom*. Taj pojam odgovara uobičajenom pojmu *dokumenta* u aplikacijama koje u jednom trenutku rade sa jednim dokumentom. *OZ Takmičenje 1.0* je aplikacija koja učitava tablu i sproviđi igru koja je definisana tablom.

OZ 1.0 je sistem od dve desktop aplikacije koje ni na koji način ne koriste Internet. Imajući u vidu da je primena veb-tehnologija kod igara sve aktuelnija, kao i činjenicu da je veb sve prisutnija tehnološka platforma u mnogim sferama, pa tako i u obrazovanju, odlučeno je da se verzija aplikacije OZ 2.0 razvija u pravcu većeg iskorišćenja veb-tehnologija, u skladu sa dugoričnijim planom orijentacije igre prema Vebu. Uočeno je da bi za kompletnu transformaciju tekuće desktop igre u veb-orijentisanu igru, osim pokretanja aplikacije sa njenog veb-sajta, trebalo rešiti sledeća tri problema. (1) Pošto je OZ obrazovna igra koja koristi table dizajnirane od strane instruktora u aplikaciji *OZ Editor*, postavlja se pitanje kako distribuirati table. Nameće se ideja da bi ove table trebalo da budu negde na Vebu, gde bi ih instruktori postavljali pošto ih naprave ili ažuriraju. (2) Kad igra više igrača, *OZ Takmičenje 1.0* je tipa vrućeg sedišta (eng. *hot seat*). Ako bi aplikacija komunicirala preko Interneta sa ostalim primercima aplikacije, igrači ne bi morali da se izmenjuju za komandama jednog računara, nego bi svaki mogao da sedi za svojim. Ovako bi se rešio i problem raspoloživih igrača, jer je lakše igraču da nađe saigrače na Internetu. (3) Kad su obrazovne igre u pitanju, za instruktore veoma je bitno da prate napredak učenika, kao i koji je učinak pojedinih tabli, pa i same igre, tj. koliko su pojedine table i igra u celini uspele da unaprede savladavanje gradiva. Mogla bi se napraviti centralizovana statistika uspeha igrača na Vebu sa kojom bi komunicirale aplikacija *OZ Editor* ili posebna aplikacija za pregled statistike koja bi mogla biti na računaru instruktora ili takođe na Vebu.

Proces okretanja igre ka Vebu započet je rešavanjem prvog problema sa gornje liste. Cilj je bio da se razviju dve funkcionalno i strukturno slične, a platformski različite varijante aplikacije, odnosno desktop aplikacija koja se instalira na računaru korisnika i aplikacija koja se može pokrenuti sa veba i

koja može preuzimati multimedijalne resurse (table i figure) iz skladišta na vebu. Za ciljnu tehnologiju je izabrana JavaFX iz razloga koji će biti objašnjeni u poglavlju 4. U ovoj fazi razvoja, samo je aplikacija *Takmičenje* konvertovana u novu JavaFX tehnologiju. Ovde treba napomenuti da JavaFX nije konvencionalna veb-tehnologija. Ona omogućava da se aplikacija instalira i pokreće kao desktop aplikacija, ali isto tako da se pokreće u čitaču veba i da sve resurse dohvata sa veba, pri čemu se ona izvršava na strani klijenta, slično konvencionalnom Java appletu. Drugim rečima, JavaFX omogućava razvoj bogate internet aplikacije (RIA), čija je celokupna „poslovna logika“ (eng. *bussines logic*) na klijentu. Iz tog razloga u ovoj tehnologiji je bilo moguće ostvariti postavljeni cilj. U strukturnom pogledu postoje samo male razlike u kodu dve ciljne varijante aplikacije i jedan mali deo, vezan za dohvatanje resursa, je implementiran pomoću veb-servisa u veb varijanti aplikacije.

Na taj način rešen je samo prvi problem, distribucija aplikacije i njenih resursa preko Veba. Rešavanjem i preostala dva problema, komunikacije između igrača i globalne statistike, još jedan deo poslovne logike bi se izvršavao na serveru i dobila bi se kompletna ciljna funkcionalnost RIA OZ. Taj deo posla uključuje više tema. (1) Rad sa korisnicima, gde bi bile najmanje tri uloge korisnika (igrač, instruktor i administrator), koji bi se prijavljivali pomoću korisničkog imena i lozinke. S tim u vezi je rešavanje raznih sigurnosnih problema, kao na primer privremeno zaključavanje naloga kod više uzastopnih unosa pogrešenih lozinki. (2) Proširenje aplikacije *OZ Editor* ili posebna veb-aplikacija za instruktore pomoću koje bi mogli da postavljaju table u skladište na Vebu i pregledaju rezultate igrača-učenika. (3) Veb aplikacija namenjena administratorima koja bi služila za kreiranje, odobravanje, otključavanje i zaključavanje naloga igrača. (4) Proširenja aplikacije *OZ Takmičenje* preko kojih bi igrači mogli da vide svoju statistiku, promene lozinku, pošalju poruku instruktoru ili administratoru sa komentarom. Iako je početni cilj projekta bio razvoj kompletne RIA, s obzirom na kompleksnost poslova vezanih za rešavanje prvog problema, odustalo se od namere da se u istom koraku (verzija OZ 2.0) reše i ostali navedeni problemi i dobije RIA sa kompletnom funkcionalnošću kombinovane desktop i veb aplikacije. Ipak, vodilo se dosta računa o modularnosti softvera i tome da se u narednim verzijama aplikacije mogu rešiti i ostali navedeni problemi bez kompleksnih izmena u postojećoj aplikaciji.

U cilju rešavanja prvog problema urađeno je sledeće. (1) Konvertovana je Java2D aplikacija *OZ Takmičenje* u JavaFX aplikaciju. Ovaj proces je oduzeo najviše vremena. (2) Urađeno je značajno refaktorisanje, kako aplikacije *OZ Takmičenje*, tako i aplikacije *OZ Editor*. (3) Razvijen je veb-servis za listanje resursa na vebu. Na njega se povezuje veb varijanta aplikacije *OZ Takmičenje* da bi dohvatila liste resursa. (4) Podržana su tri načina izvršavanja aplikacije *OZ Takmičenje*: (a) kao konvencionalne desktop aplikacije, (b) kao RIA koja se izvršava u okviru prozora čitača veba i (c) kao aplikacije koja se preko čitača veba pokreće preko hiper-linka na veb-stranici, a zatim izvršava u posebnom prozoru aplikacije (kao i desktop aplikacija) i sopstvenom procesu operativnog sistema. Ova poslednja mogućnost je realizovana zahvaljujući *Java Web Start* tehnicima.

3. OBRAZOVANJE KROZ ZABAVU

Godine 1974. Ričard Djuk (*Richard Duke*), profesor na Univerzitetu Mičigen, sugerisao je da računarske igre treba da postanu standardno sredstvo u obrazovanju [10]. Djuk je bio pionir u istraživanju primene računarskih igara u obrazovanju. Od njegovog vremena pa do danas je objavljeno mnogo rezultata istraživanja iz ove oblasti, a pojavio se i veliki broj obrazovnih igara. U [11] je zapaženo da se često proizvode igre prototipovi koji nisu javno dostupni i koji nisu formalno testirani, kao i to da se objavljuju naučni radovi koji istražuju nove pristupe, ali da se sve završava na tome. U [12] stoji da se 2007. godine istraživanjem na polju obrazovnih igara bavilo preko 190 institucija u SAD i još 161 institucija van SAD.

Prema Hortonu [13] igra se posmatra kao aktivnost koja se može dizajnirati tako da bude test, lekcija ili ceo kurs. Takođe, [13] sadrži preporuke i uputstva kako da se oblikuje jedna obrazovna igra. Glavna ideja je da se prvo utvrdi tačno šta je cilj igre, koja je tema igre, odnosno koju oblast igra treba da podučava. A zatim, kada su tema i cilj definisani, razvoj igre na datu temu treba da bude određen datim ciljem. Po Hortonovim preporukama igra se može posmatrati kao test koji daje neposredan odziv na akcije igrača i koji ocenu uspeha datu igraču ne koristi kao zvaničan rezultat – na taj način igrač ima slobodu da nagađa i isprobava alternativne pristupe. Na ovom mestu treba naglasiti da igra OZ, o kojoj je reč u ovom radu, suštinski predstavlja fleksibilno okruženje za projektovanje i sprovođenje igara sa različitim tematikom, te u tom smislu ne prati direktno Hortonovo uputstvo vezano za oblast i temu konkretne igre. Upravo je reč je o softveru koji je potpuno fleksibilan i ni na koji način ne zavisi od tematike igre. Oblast i temu testiranja znanja određuje korisnik – instruktor kada formira kviz, bez programiranja, korišćenjem apstrakcija visokog nivoa koje mu pruža razvijeni softver. Tek prilikom projektovanja konkretnih kvizova (igara) koristeći OZ, instruktor treba da vodi računa o Hortonovim uputstvima.

U [11] se navodi istraživanje koje je uzelo u obzir 200 radova vezanih za obrazovanje kroz zabavu, objavljenih u periodu od 2000. do 2004. godine i koje je došlo do zaključka da su u većini slučajeva razvijene simulacije tako da je podučavani učio iz svojih iskustava. Iako u literaturi postoji neslaganje oko toga da li je simulacija igra, u [13] se napominje da treba razlikovati simulaciju od demonstracije, jer u ovoj drugoj korisnik ne kontroliše sled događaja. U ovom tekstu će se simulacija smatrati posebnim žanrom igre, u koji ne spada OZ.

Prema [14], među prvim komercijalnim obrazovnim igrama je *Number Munchers* iz 1980. godine. Ova igra namenjena je uvežbavanju osnovnih matematičkih operacija. Sredinom 90-tih igra se pojavljuje pod novim imenom *Math Munchers*. U tabeli 1, preuzetoj iz [15], prikazani su primeri aktuelnih uspešnih komercijalnih obrazovnih igara, organizovanih po oblastima kojima se bave. Osim komercijalnih, mogu se naći i besplatne obrazovne igre; u tabeli 2 su navedene lokacije na Vebu koje predstavljaju agregatore besplatnih obrazovnih igara.

Igre se razvijaju i na univerzitetima i drugim institucijama kojima nije u fokusu komercijalna dobit i to najviše u SAD. Kod igara varira uzrast kome su namenjene, oblast kojoj podučavaju,

kao i sam žanr igre. Na Karnegi Melon Univerzitetu razvijena je igra *Alice* namenjena studentima tog univerziteta, zamišljena tako da u 3D okruženju studenti uče da programiraju [16]. Na Državnom Univerzitetu Severne Dakote razvijeno je virtuelno okruženje *Geology Explorer* namenjeno studentima geologije da usavršavaju svoje veštine istraživanja i zaključivanja na osnovu prikupljenih podataka [17]. U [18] istražuju se mogućnosti da se igre koriste za učenje stranih kultura i jezika i prikazuje se nekoliko takvih igara. U [19] se istražuje obrazovni potencijal igara u kategoriji MMORPG (eng. *Massively Multiplayer Online Role-Playing Game*).

Olimpijada znanja 1.0 je opisana u [5, 6]. U pitanju je potezna igra sa linearnom otvorenom putanjom kojom se napreduje bacanjem kocke. Igra je za 1-8 igrača tipa „vrućeg sedišta“. Stajanjem na neka polja igraču se postavlja pitanje i, u slučaju da ne odgovori tačno, biva kažnjen vraćanjem nazad na prethodno polje, početno polje ili neko drugo unapred utvrđeno polje. Igra se može koristiti za proizvoljan uzrast i proizvoljnu oblast. *Olimpijada znanja 2.0*, predmet ovog rada, je funkcionalno praktično istovetna prethodnoj verziji, s tom razlikom da se može igrati pokretanjem i korišćenjem resursa sa servera na Internetu.



Slika 1: Aplikacija *Takmičenje*. Igra za 3 igrača. Igrač je na potezu i čeka se da klikne na kocku

Tabela 1: Primeri uspešnih komercijalnih obrazovnih igara

Oblast	Naslov igre
Ekologija	Operation: Resilient Planet
Opšte nauke	Science Pirates: The Curse of Brownbeard
Istorija i političke nauke	Conspiracy Code Discover Babylon Making History Peacemaker Global Conflicts
Medicina	Re-Mission 2
Matematika	DimensionM Lure of the Labyrinth
Organizacione nauke	Virtual U
Mašinstvo	Time Engineers
Mrežne tehnologije	Cisco Mind Share
Fizika	Crayon Physics Physicus Coaster Creator Phun

Tabela 2: Veb lokacije sa besplatnim obrazovnim igrama

Naziv veb lokacije	Veb adresa
Educational Freeware	http://www.educational-freeware.com/
Social Impact Games	http://www.socialimpactgames.com

Igra koja ima sličnosti sa OZ je *TRIRACE* [20]. Njena namena je podučavanje matematike u osnovnim školama. Ova igra je adaptacija stone igre i ima otvorenu linearnu putanju sa poljima popunjenim raznim trouglovima. Jedan ili dva igrača napreduju po putanji tako što dobijaju karte sa zadacima da pronađu trougao sa zadatim karakteristikama. Ako pronađu odgovarajući trougao idu na polje gde se pronađeni trougao nalazi. Ako pogreše ili ne mogu da pronađu trougao ostaju na polju na kome su bili. Kao faktor iznenađenja igri su dodate još dve karte koje ne sadrže pitanje: ako izvuče prvu – igrač se vraća na početno polje, a ako izvuče drugu – igrač preskače jedan potez. Igra je, dakle, poteznog tipa i igra se za jednim računom.

Još jedna igra koja ima sličnosti sa OZ je *STIs: Stopping the transmission* [21]. U pitanju je adaptacija američke stone igre *Parčizi* (eng. *parcheesi*) koja podseća na dobro poznatu igru *Ne ljuti se, čoveče*. Igra je tipa otvorene linearne putanje. Namenjena je adolescentima, uzrasta od 14 do 15 godina, sa ciljem da ih poduči o polno prenosivim bolestima. Igra je za 1-4 igrača i svakom igraču su dodeljene po 4 figure i po jedna oblast. Potez počinje tako što se igraču postavlja pitanje iz oblasti koja mu je dodeljena i, ako tačno odgovori na pitanje, igrač može da baci kocke i da odigra potez nekom od figura u skladu sa ishodom na kockama. Igra ima sporu i brzu putanju i početna polja za igrače su polja njihovog ukrštanja. Kada se figura nađe ili samo pređe preko početnog polja, ako na prvo sledeće pitanje igrač odgovori tačno, figura nastavlja po brzoj putanji, u suprotnom figura nastavlja po sporoj putanji. Vreme za odgovore je ograničeno. Igra se može igrati za jednim računom u „vrućem sedištu“ ili u mrežnom režimu.

U tabeli 3 data je uporedna analiza OZ i dve opisane igre koje imaju određene sličnosti sa OZ.

4. ANALIZA TEHNOLOGIJA

Danas se koristi veliki broj tehnologija za izradu veb aplikacija. Ove tehnologije se mogu podeliti u nekoliko glavnih grupa prema njihovoj nameni: osnovne veb tehnologije (HTML, CSS, XML, JSON, ...), tehnologije za razvoj aplikacija na serveru (ASP.NET, ASP.NET MVC, JSP, JSF, Java Servlet, PHP, Ruby on Rails, Coldfusion, ...) i tehnologije za razvoj aplikacija na klijentu (JavaScript, Java Applet, JavaFX, Silverlight, Flex, ...). Aplikacije na serveru mogu koristiti i neku bazu podataka (SQL Server, MySQL, ...), kao i neku od modernih tehnologija koje podržavaju perzistenciju podataka (Spring, Hibernate, Entity Framework, ...). Druga podela tehnologija je prema platformi na kojoj su zasnovane. Po zastupljenosti se izdvajaju tehnologije zasnovane na Microsoft .NET platformi (ASP.NET, ASP.NET MVC, Silverlight, ...), u kom slučaju se najčešće koriste jezici C# i Visual Basic, one zasnovane na Java platformi (JSP, JSF, Java Servlet, Java Applet, JavaFX, ...) gde se najčešće koristi jezik Java i one zasnovane na Adobe Flex platformi, gde se koristi jezik ActionScript. Nije neophodno da tehnologije za serverske i klijentske aplikacije budu na istoj platformi, jer se klijentska i serverska aplikacija izvršavaju nezavisno, a veza između njih se ostvaruje međusobnom komunikacijom. Ali ako su obe ove tehnologije na istoj platformi pojednostavljuje se razvoj, najpre zbog toga što se može koristiti isto razvojno okruženje.

Konvencionalna veb aplikacija, ili tanki klijent kako se još naziva, ima straničnu strukturu, izvršava se u čitaču veba, ne zahteva instalaciju, ažuriranje aplikacije je nevidljivo za korisnike i korisnik može da pokrene aplikaciju sa bilo kog uređaja koji ima samo čitač veba, bez posebnih izvršnih okruženja i dodataka. Desktop aplikacija mora da se instalira i održava, ali može da poseduje bogat korisnički interfejs. Bogata internet aplikacija (RIA), ili kako se ponekad naziva „debeli klijent“, je aplikacija koja kombinuje prednosti konvencionalne veb i desktop aplikacije. Neki autori nazivaju debelim klijentom i desktop aplikacije koje komuniciraju sa serverom na Internetu.

Tabela 3: Poređenje karakteristika tri međusobno slične obrazovne igre

	TRIRACE	STIs: Stopping the transmission	Olimpijada znanja
Način napredovanja po putanji	Izvlačenje karte, pozicija trougla opisanog na karti i prepoznavanje trougla	Ishod na 2 kocke i uspeh odgovora na pitanje	Ishod na kocki i uspeh odgovora na pitanje određeno poljem na koje se staje
Tipovi pitanja	Pronalaženje trougla sa navedenim karakteristikama	Unos odgovora, izbor odgovora, sekvence, neka pitanja dozvoljavaju dodavanje slika, zvuka i videa u odgovor	Unos odgovora, izbor odgovora, izbor slike, pogađanje slova
Prateća multimedija uz pitanja	Nema	Slika, zvuk i video	Slika i zvuk
Tipovi kazni	Vraćanje na prethodno polje, vraćanje na početak, preskakanje poteza	Nenapredovanje, sporo napredovanje, vraćanje na početak	Vraćanje na prethodno polje, vraćanje na tačno određeno polje, vraćanje na početak
Kada se igrač kažnjava	Ako ne može da pronađe trougao opisan na karti ili izvuče kaznenu kartu	Ako ne uspe da tačno odgovori na pitanje ili ako figura drugog igrača stane na isto polje	Ako ne uspe da tačno odgovori na pitanje
Oblast podučavanja	Trouglovi	Seksualno prenosive bolesti	Može biti bilo koja oblast
Ciljani uzrast	Osnovna škola	14-15 godina	Može biti bilo koji uzrast
Broj igrača	1-2	1-4	1-8
Broj figura po igraču	1	4	1
Igranje preko mreže	Ne	Da	Ne

Moderne tehnologije koje se oslanjaju isključivo na čitače veba su HTML 5, CSS i JavaScript. Postoji i veliki broj JavaScript biblioteka koje se koriste za izradu vizuelnih komponenti na veb stranama, kao što je jQuery. Ajax je tehnologija koja obezbeđuje da se razmenjuju asinhronne poruke između klijenta i servera u XML (*Extensible Markup Language*) formatu i da se odziv sa servera koristi za dinamičku manipulaciju nad delom sadržaja veb strane. Za razmenu poruka i manipulaciju nad veb stranom na strani klijenta koristi se jezik JavaScript. Ovim je postignuto da se veb strana dinamički menja bez potrebe za njenim ponovnim učitavanjem. Pažljivo osmišljavanje kada i kako će se menjati veb strana može značajno da doprinese doživljaju korisnika. Tehnologije kao što je WebGL omogućavaju i razvoj klijentske strane sa naprednom 2D ili 3D grafikom. Iako je moguće kreirati aplikacije sa bogatim skupom komponentata i bogatom grafikom, ovakve aplikacije su još uvek klasične veb aplikacije (tanki klijenti).

Za bogate internet aplikacije danas se koriste tehnologije koje se oslanjaju na posebna izvršna okruženja, od kojih su neka dodaci (eng. *plug-in*) čitača veba. Sa druge strane, ovakve aplikacije ne zavise od mogućnosti samog čitača. Danas najrasprostranjenije tehnologije su Adobe Flex, Oracle JavaFX i Microsoft Silverlight. Dobra strana ovakvih tehnologija je izbegavanje različitog ponašanja kakvo ispoljavaju klasične veb aplikacije u različitim čitačima veba. U slučaju RIA koja se izvršava na dodatku, ponašanje aplikacije je u potpunosti pod kontrolom dodatka, tako da će u svim čitačima veba biti isto. Loša strana ovoga je da možda ne postoje adekvatni dodaci za sve čitače veba. Ipak, za tri glavne RIA tehnologije, dodacima je obezbeđena podrška za najveći broj modernih čitača veba.

Navedene tri najčešće korišćene RIA tehnologije su veoma slične među sobom i u tabeli 4 su date neke njihove karakteristike. Sve tri tehnologije imaju izvesnu podršku za slike, audio i video datoteke i uglavnom se razlikuju u formatima koje podržavaju. Aplikacije razvijene u ovim tehnologijama mogu da se izvršavaju kao desktop, veb ili mobilne aplikacije. Silverlight i JavaFX podržavaju pokretanje aplikacije iz čitača veba, a zatim njeno izvršavanje kao desktop aplikacije. Sve tri tehnologije pružaju mogućnost programiranja na deklarativnom jeziku za-

snovanom na XML, kao i na objektno orijentisanom programskom jeziku. Sve što se opiše deklarativnim jezikom se može opisati i programskim jezikom i stvar je izbora programera, na kraju se i deklarativni i programski opis prevodi u isti format i koristi iste biblioteke klasa. Flex i JavaFX podržavaju CSS za stilizaciju grafičkog korisničkog interfejsa.

JavaFX je najmlađa od ove tri RIA tehnologije. Kad se pojavila, JavaFX 1.0 nije koristila Java jezik već JavaFX Script jezik koji je bio deklarativan jezik nalik na jezik Java. JavaFX 2.0 se pojavila 2011. godine i unela je velike izmene u ovu tehnologiju – JavaFX Script je napušten i umesto toga uvedeni su deklarativni jezik FXML i jezik Java. Verzija JavaFX 2.2 je postala sastavni deo Java platforme. Takođe, popularna razvojna okruženja za Javu, kakava su NetBeans i Eclipse, su uključila podršku i za JavaFX. Jedan od korisnih efekata ovih promena je omogućena integracija Java i JavaFX aplikacija.

OZ 1.0 je razvijena na Java platformi pri čemu su korišćene tehnologije Swing i Java2D. U ovom radu smo se fokusirali na konverziju aplikaciju *Takmičenje* radi njene orijentacije ka vebu. *Editor* je po prirodi tipična desktop aplikacija, namenjena mnogo manjem broju korisnika od *Takmičenja*, te nije bilo potrebe za značajnijim izmenama. Veb varijanta aplikacije *Takmičenje* bi se sastojala od dve različite komponente: one koja se izvršava na klijentu i one koja se izvršava na serveru. Zbog toga je bilo potrebno napraviti izbor dve tehnologije: klijentske i serverske.

S obzirom na to da su aplikacije OZ 1.0 razvijene na jeziku Java, posebno su bile analizirane dve mogućnosti: (1) da se nastavi razvoj u istim tehnologijama u kojima je razvijena OZ 1.0 (Java2D i Swing) ili (2) da se pređe na moderniju tehnologiju (JavaFX). Zadržavanjem iste tehnologije bio bi izbegnut relativno složen i vremenski zahtevan posao konverzije u novu tehnologiju, ali sledeći razlozi su prevagnuli u odluci da se u razvoju OZ 2.0 pređe na novu tehnologiju JavaFX:

1. Atraktivniji izgled JavaFX od Swing GUI kontrola. Pri tome JavaFX sadrži gotovo sve GUI kontrole koje sadrži Swing, kao i mogućnosti crtanja koje ima Java2D. JavaFX se može posmatrati kao biblioteka koja je objedinila Swing i Java2D na jedan moderan način.

Tabela 4: Usporedna tabela 3 najčešće korišćene RIA tehnologije

	Adobe Flex	Microsoft Silverlight	Oracle JavaFX
Godina prve verzije	2004.	2007.	2008.
Razvojna okruženja	Adobe Flash Builder, Adobe Flash Catalyst, Adobe Flash Professional, FlashDevelop	Microsoft Visual Studio, Microsoft Expression Blend	NetBeans, Eclipse
Deklarativni jezik	MXML	XAML	FXML od verzije 2.0
Programski jezik	ActionScript	C#, Visual Basic	Java od verzije 2.0
Podrška za CSS	Da	Ne	Da
Podrška za 3D grafiku	Obezbeđena preko eksternih biblioteka	XNA 3D API od verzije 5	JavaFX 3D od verzije Java 8 koja je najavljena za početak 2014.
Ekstenzija prevedene aplikacije	SWF	XAP	JAR
Dodatak	Adobe Flash Player za čitače veba i Adobe AIR za desktop	Silverlight	Java 7 Update 6 ili novija verzija
Ugrađena podrška da se aplikacija pokrene iz čitača veba, a zatim izvršava kao desktop aplikacija	Nema	Out Of Browser	Web Start

2. Jednostavniji rad sa grafikom. Očekuje se da će unapređenje grafike u budućim verzijama aplikacije zahtevati manje programerskog truda i vremena.
3. Dostupne biblioteke za jednostavan rad sa animacijom. Dodavanjem vizuelnih efekata u budućim verzijama aplikacije može se dobiti na vizuelnom doživljaju korisnika.
4. Odlična audio podrška. JavaFX ima ugrađenu podršku za zvuk, tako da se ne moraju koristiti eksterne biblioteke.
5. Video podrška koja trenutno obuhvata samo par formata, ali se očekuje da će u budućnosti biti podržano više formata. Mogla bi korisno da posluži u nekoj budućoj verziji aplikacije, za pitanja ili odgovore u obliku video sekvenci.
6. Mogućnost da se u istoj tehnologiji napravi desktop i veb (RIA) varijanta iste igre i da se varijante veoma malo razlikuju u izvornom kodu. A zatim mogućnost da se veb varijanta aplikacije pokrenuta iz čitača veba izvršava u prozoru čitača, ili u vlastitom prozoru, vizuelno identično kao da je u pitanju desktop aplikacija. Iako Java Web Start tehnika omogućava da se i aplikacija razvijena u Java2D tehnologiji pokrene sa Veba, ne bi bilo moguće da se isti Java2D program pokreće u prozoru čitača veba ili u vlastitom prozoru i vlastitom procesu: za program koji se izvršava u prozoru i procesu čitača veba morao bi se pisati aplet, a za aplikaciju koja se izvršava u vlastitom prozoru i procesu, morala bi se pisati konvencionalna aplikacija u Java2D tehnologiji.
7. Opštepoznata činjenica da modernijoj tehnologiji proizvođači razvojnih alata i biblioteka poklanjaju više pažnje, te će održavanje softvera i razvoj narednih verzija profitirati na toj činjenici.
8. Programski jezik i razvojno okruženje se ne bi menjali i očekivano je bilo da će veliki deo izvornog koda biti iskorišćen.

Java platforma je izabrana i za serversku tehnologiju. Trebalo je izaberati tehniku kojom će se dobijati liste datoteka dostupnih na serveru za figure igrača i table za igru. Postojalo je više mogućnosti: JSP (*Java Server Pages*), Java Servlet ili veb servis. Odlučeno je da se koristi veb servis implementiran pomoću JAX-WS (*Java API for XML Web Services*), jer je to omogućilo da slanje podataka preko mreže bude potpuno transparentno za programera – sa obe strane (klijentske i serverske) se radi sa Java objektima. Na taj način programer ne brine o WSDL (*Web Services Description Language*) opisu (datoteka koja sadrži opis veb servisa) niti o strukturi JSON (*JavaScript Object Notation*) ili XML datoteka koje bi se slale preko mreže i sadržale podatke sa servera.

Suštinska razlika na klijentskoj strani, između prethodno korišćene Java2D tehnologije i JavaFX tehnologije korišćene u novoj verziji aplikacije, je u načinu crtanja. Tipično crtanje na panelu u tehnologiji Java2D, pomoću klasa iz Swing paketa, se radi na sledeći način: programer kreira klasu panela tako što nasledi `javax.swing.JPanel` i u toj klasi redefiniše (nadjačava) metodu za crtanje na sledeći način:

```
@Override
public void paint(java.awt.Graphics g) {
    super.paint(g);
    Graphics2D g2d = (Graphics2D) g;
    // kod za crtanje
}
```

gde objekat klase `Graphics2D` predstavlja kontekst za crtanje. Unutar ove metode programer piše kod koji iscrtava kompletni sadržaj na panelu, pri čemu je redosled bitan i važi pravilo da poslednje iscrtani sadržaj prekriva prethodno iscrtani. Iscrtavanje se vrši pozivanjem metoda nad objektom klase `Graphics2D`. Metodu za crtanje programer po pravilu ne zove eksplicitno, već nju po potrebi poziva GUI nit sistema. Programer može da pozove metodu kojom se sistemu šalje poruka da treba ponovo da iscrtava sadržaj panela. Unutar klase panela biće jedan ili više atributa koje čine stanje panela. Atributi stanja panela mogu biti primitivni podaci (logički, brojevi ili tekstualni) ili objekti za crtanje (linija, pravougaonik, elipsa). Programer ažurira stanje panela i šalje preporuku za ponovno iscrtavanje, dok sistem poziva metodu za crtanje koja zatim na osnovu stanja panela iscrtava njegov sadržaj.

Tipično crtanje na panelu u tehnologiji JavaFX se radi na sledeći način: programer kreira klasu panela tako što nasledi `javafx.scene.layout.Pane`. Objekat panela sadrži jedan ili više čvorova, objekata klase `javafx.scene.Node`, koji čine njegov sadržaj za iscrtavanje. U JavaFX terminologiji čvorovi koji su postali sadržaj panela nazivaju se njegovom `decom` i čuvaju se u listi tipa `ObservableList<Node>`. Sadržaj panela se gradi dodavanjem čvorova tako što se prvo dohvati lista (eventualno već postojeće) dece panela, a zatim se na ovu listu doda novi čvor, koji tako postaje dete panela. Ne postoje metode za crtanje ili preporuke za ponovno iscrtavanje, već panel nadgleda listu svoje dece i reaguje na svako dodavanje ili brisanje čvora iz liste ili promenu atributa bilo kog čvora iz liste ponovnim iscrtavanjem panela. U sledećem primeru se kreira kvadrat stranice 100 tačaka i dodaje panelu (`Rectangle` nasleđuje `Node`), zatim se menja veličina stranice kvadrata, a na kraju se kvadrat uklanja sa panela:

```
Rectangle rect = new Rectangle();
rect.setWidth(100);
rect.setHeight(100);
getChildren().add(rect);
...
rect.setWidth(200);
rect.setHeight(200);
...
getChildren().remove(rect);
```

Redosled dodavanja dece određuje redosled iscrtavanja, tako da kasnije dodato dete može prekriti prethodno dodato. Na ovaj redosled se može uticati pozivanjem metode nad detetom koja će ga staviti u prednji plan, odnosno u zadnji plan.

JavaFX je obezbedila bogatu biblioteku klasa koje nasleđuju Node: geometrijski oblici, slika, video, GUI kontrole, pa čak i čitač veba (koristi se *WebKit*). Pane takođe nasleđuje Node, tako da se na ovaj način sadržaju panela može dodati drugi panel, odnosno formirati njihova kompozicija proizvoljne dubine.

Razlika u načinu formiranja, ažuriranja i iscrtavanja scene je suštinska, posmatrano iz ugla programera. U slučaju Java2D za inicijalno crtanje se ne može koristiti konstruktor klase panela i moraju se čuvati reference na sve objekte koji se crtaju, makar se nikad ne promenili. Kod JavaFX inicijalni ili neopromenljivi sadržaj se može izgraditi u konstruktoru i nije neophodno čuvati referencu na objekat koji se crta ukoliko znamo da se neće menjati. U JavaFX čak nije potrebno naslediti Pane da bi se na njemu crtalo, može se kreirati jedan objekat Pane i izgraditi njegov sadržaj dodavanjem čvorova, jer je metoda za dohvatanje deca javna. U slučaju Java2D aplikacije, s jedne strane programer ima u vidu stanje panela koji on ažurira na osnovu raznih događaja, a s druge strane programer obezbeđuje metodu za crtanje koja crta sadržaj panela na osnovu stanja panela. U slučaju JavaFX aplikacije, programer treba da izgradi inicijalni sadržaj panela, a zatim na osnovu raznih događaja da samo menja sadržaj panela, odnosno dodaje, uklanja i ažurira njegovu decu (objekte koji se crtaju), dok o njihovom iscrtavanju ne brine.

5. PROJEKAT I IMPLEMENTACIJA SOFTVERA

Kako je prethodno napomenuto, da bi se postigao cilj razvoja dve funkcionalno i strukturno malo različite varijante (za desktop i veb) obrazovne igre OZ, zbog niza navedenih razloga u prethodnom odeljku, odlučeno je da se aplikacija *Takmičenje* konvertuje u novu i modernu tehnologiju, JavaFX. Iako se način formiranja i prikazivanja scene bitno razlikuje između tehnologija Java2D i JavaFX, klase iz paketa Swing uglavnom imaju svoje ekvivalente u paketima JavaFX. U ovom odeljku su objašnjene napravljene izmene i način na koji su izmene sprovedene.

Klase koje nasleđuju `javax.swing.JPanel` su konvertovane u sledeće klase iz paketa `javafx.scene.layout`: Pane, GridPane, VBox ili HBox. Dok se kod Swing-a raspoređivanje kontrola (eng. *layout*) uobičajeno definiše u konstruktoru klase, kod JavaFX način raspoređivanja kontrola je unapred definisan potklasom Pane koja se koristi. Konvertovanje klase Igra iz naslednika JPanel u naslednika Pane, najveće klase kako u OZ 1.0 tako i u OZ 2.0, zahtevalo je veći programerski napor i vreme, zbog razlika u načinu crtanja koja je objašnjena u poglavlju 4. Pošto JavaFX nema klasu koja odgovara `javax.swing.JDialog`, razvijena je klasa Report koja nasleđuje `javafx.scene.Stage`, ekvivalent klasi `javax.swing.JFrame`.

Klase geometrijskih figura iz paketa Swing zamenjene su svojim ekvivalentima u JavaFX. Klase GUI kontrola zamenjene su takođe svojim ekvivalentima, osim u slučaju `javax.swing.JSpinner` koja nema svog ekvivalenta, pa je zamenjena sa `javafx.scene.control.ComboBox`. Za izbor datoteke i foldera, umesto `javax.swing.JFileChooser`, JavaFX je obezbedila

dve posebne klase, jednu za izbor datoteka `javafx.stage.FileChooser` i jednu za izbor foldera `javafx.stage.DirectoryChooser`.

Rad sa slikama značajno je pojednostavljen u JavaFX u odnosu na Swing. Koriste se dve posebne klase: `javafx.scene.image.Image` koja služi da se učita slika iz datoteke, dok `javafx.scene.image.ImageView` služi da se prikaže slika. Objekti klase `ImageView` se dodaju na objekat panela Pane da bi slika postala sastavni deo panela i tako se prikazala u panelu. Za reprodukciju zvučnih zapisa umesto `javax.sound.sampled.Clip` prešlo se na `javafx.scene.media.AudioClip`. Eksterna biblioteka za reprodukciju MP3 datoteka, koju aplikacija *Editor* i dalje koristi, zamenjena je u aplikaciji *Takmičenje* sa `javafx.scene.media.MediaPlayer`.

Klasa `IconManager`, koja se u OZ 1.0 koristi za dohvatanje figura igrača, je proširena (i preimenovana u `Resources`) i u OZ 2.0 se koristi za dohvatanje svih resursa (slika i zvuka), u aplikaciji *Takmičenje*. Ova klasa vodi računa o tome da li su resursi na lokalnom računaru ili na serveru, odnosno da li se radi o desktop ili veb varijanti aplikacije.

Pošto *Editor* i dalje koristi Swing, table za igru sadrže boju sačuvanu kao jedan celobrojni podatak koji se zatim koristi u konstruktoru tipa `java.awt.Color`. Međutim JavaFX za boje koristi `javafx.scene.paint.Color` i objekat ove klase ne može da iskoristi podatak o boji koji je sačuvala *Editor*. Zbog toga je napisana javna statička metoda koja konvertuje objekat AWT boje u objekat JavaFX boje.

Aplikacija *Takmičenje* je razvijana na Java 7 platformi, jer je JavaFX uvedena tek u toj verziji platforme. Aplikacija *Editor* je mogla ostati na starijoj verziji Jave, ali je odlučeno da se i ona prenese na Java 7 platformu, kako bi se obe aplikacije prevodile istom verzijom prevodioca. Ovo je omogućilo da se i u *Editoru* koriste elementi jezika Java uvedeni tek u ovu verziju, uz refaktorisanje koda manjeg obima. Ti novi elementi su *try-with-resources* [22], *diamond inference* [23] i *multi-catch* [24]. Ipak, aplikacija *Editor* nije konvertovana u JavaFX tehnologiju, već se i dalje njen korisnički interfejs oslanja na paket Swing.

Osim konverzije i refaktorisanja aplikacija, bilo je potrebno prebaciti resurse koje koristi aplikacija *Takmičenje* (ikone, table) na Veb. Dok desktop varijanta radi sa resursima na lokalnom računaru, veb varijanta dohvata ove resurse sa servera. Razlike između ove dve varijante u funkcionalnom smislu praktično ne postoje. Razlika je u tome da veb varijanta zahteva internet vezu i da se može pokrenuti sa servera i izvršavati unutar prozora čitača veba ili u vlastitom prozoru i procesu. Desktop i veb varijanta aplikacije *Takmičenje* se međusobno razlikuju u nekoliko desetina linija koda, a za odlučivanje o kojoj varijanti aplikacije se radi, koristi se jedna `public static final boolean` konstanta. Kako je konstanta poznata u vreme prevodjenja prevodilac će isključiti deo koda koji ne može da se dohvati i neće ni testirati ovu konstantu u vreme izvršavanja – na ovaj način je postignuto da su obe varijante aplikacije jednako efikasne kao da su posebno pisane za desktop, odnosno veb.

Za dohvaćanje resursa koji se mogu menjati, figure igrača i table za igru, uveden je veb servis. Aplikacija *Takmičenje* poziva veb servis sa zahtevom za liste figura ili tabli, a odziv servisa je lista resursa na Vebu. Dalje *Takmičenje* može da preuzima pojedinačne resurse koristeći ove liste. Za razvoj veb servisa korišćena je tehnologija JAX-WS. Ova tehnologija omogućava programeru jednostavan razvoj veb servisa uz pomoć anotacija – dovoljno je napisati Java klasu i anotacijama naznačiti bitne stavke za veb servis i JAX-WS će generisati samu aplikaciju veb servisa i njegov WSDL opis. Takođe, ova tehnologija omogućava jednostavno pozivanje veb servisa tako što na osnovu WSDL opisa generiše klase koje zatim programer koristi da bi pozivao veb servis. Ceo proces je veoma jednostavan i programer ne mora da se upušta u detalje mehanizma veb servisa, čak ne mora ni da brine o sadržaju WSDL opisa.

WSDL opis, osim što sadrži podatke o tome kako pozvati i kako tumačiti rezultate poziva veb servisa, sadrži i veb adresu na kojoj se veb servis nalazi. U toku razvoja se ne zna na kojoj će adresi biti veb servis koji će koristiti veb varijanta aplikacije *Takmičenje*, ali se zato zna da će aplikacija koja ga poziva, urađena u JavaFX tehnologiji, biti na istom serveru kao i veb servis. Kako JAX-WS obezbeđuje metode za pozivanje veb servisa na proizvoljnoj adresi, ignoriše se veb adresa iz WSDL opisa i veb servis se poziva pomoću veb adrese određene na osnovu one sa koje je učitana aplikacija *Takmičenje*. Svaka JavaFX aplikacija mora da ima jednu klasu koja nasleđuje `javafx.application.Application` čija metoda `public static void main(String[] args)` predstavlja ulaznu tačku aplikacije. Kada je u pitanju *Takmičenje*, ta klasa se naziva `TakmicenjeFX` i ako je app referenca na objekat te klase, tada se veb lokacija sa koje je pokrenuto *Takmičenje* dobija na sledeći način:

```
app.getHostServices().getCodeBase()
```

Ovim je postignuto da se obe komponente aplikacije *Takmičenje* (aplikacija koja se preuzima sa servera a izvršava na klijentu i veb servis) mogu postaviti na bilo koji server. Mana ovog pristupa je da WSDL opis sadrži netačnu adresu veb servisa. Čak i ako se obezbedi da WSDL opis sadrži tačnu veb adresu, jedino što bi se moglo dobiti je da se veb servis jednostavno može zvati od strane drugih aplikacija i drugih servera, što ovde nije bio cilj, jer je ovaj veb servis razvijen isključivo za potrebe igre *Takmičenje*.

Razvoj OZ 2.0 je počeo na Windows 7, a završio na Windows 8 operativnom sistemu. Uvek je korišćena najnovija dostupna verzija razvojnog okruženja NetBeans IDE, a u vreme završetka projekta to je bila verzija 7.3.1. Za prevođenje je uvek korišćena najnovija verzija Java platforme, a konačna verzija aplikacije je prevedena u verziji Java 7 Update 25. Veb varijanta aplikacije *Takmičenje* je testirana na aplikativnom serveru GlassFish, verzije 3.1.2 i 4.0. Ponašanje aplikacije na klijentu testirano na čitačima veba koji su u masovnoj upotrebi (Mozilla Firefox, Microsoft Internet Explorer, Google Chrome).

Datoteke slika u zvučne datoteke u OZ 2.0 nisu menjane u odnosu na prethodnu verziju. Sve datoteke slika su u PNG formatu, dok su sve zvučne datoteke u WAV formatu. Slike u aplikacijama *Editor* i *Takmičenje* su preuzete iz *Open Icon Library* [25]. Figure igrača u aplikaciji *Takmičenje* su preuzete iz baze besplatnih ikonica firme *Iconshock*. Zvukovi su preuzeti iz biblioteke zvukova *One Laptop Per Child* [26]. Korišćene su sledeće eksterne biblioteke van standardnog paketa biblioteke Java 7 platforme: (1) *JLayer 1.0.1* – koristi aplikacija *Editor* za reprodukciju MP3 datoteka; i (2) *Apache Commons IO 2.4* – koristi aplikacija *Takmičenje* za preuzimanje datoteka tabli sa servera.

6. ZAKLJUČAK

U vreme kada je započet razvoj prve verzije obrazovne igre *Olimpijada znanja*, koju čine dve aplikacije *Editor* – za instruktore koji pripremaju table za igru i odgovarajuće kvizove znanja, i *Takmičenje* – za učenike-igrače, tehnologija JavaFX još nije postojala, tako da su aplikacije u prvoj verziji realizovane u Java2D tehnologiji. Međutim, u trenutku kada se odlučivalo o tehnologiji za razvoj nove verzije *Olimpijade znanja* orijentisane prema Vebu, JavaFX je već postala dovoljno zrela tehnologija, pogodna za razvoj bogatih internet aplikacija. Imajući u vidu cilj uporednog razvoja varijanti aplikacije za desktop i veb, aplikacija *Takmičenje* OZ 2.0 je u potpunosti razvijena u ovoj modernoj tehnologiji.

Dok je aplikacija *Takmičenje* konvertovana u tehnologiju JavaFX, aplikacija *Editor* je samo refaktorisana onoliko koliko je to bilo neophodno, shodno promenama koje je pretrpela aplikacija *Takmičenje*. Aplikacija *Takmičenje* je razvijena u dve varijante, koje se veoma malo razlikuju u izvornom kodu. Desktop varijanta se instalira na računaru zajedno sa svim resursima i njoj nije potrebna internet konekcija. Veb varijanta se pokreće iz čitača veba i to na dva načina – unutar samog čitača ili kao *web start* – u posebnom prozoru i procesu, kao nezavisna aplikacija u odnosu na čitač. Bilo kako da je pokrenuta, veb varijanta aplikacije ima sve resurse na vebu i potrebna je internet veza za njeno korišćenje.

Kako je osnovni cilj ovog rada bio prenos aplikacije *Takmičenje* iz Java2D u JavaFX tehnologiju sa idejom njenog kasnijeg unapređenja u aplikaciju potpuno orijentisanu ka Vebu, u radu su ispitane samo obe mogućnosti pokretanja nove verzije aplikacije sa veba i njenog povezivanja sa odgovarajućim veb servisom za pristup resursima na serveru. U tekućoj verziji 2.0 nisu implementirane sve funkcionalnosti kompletne veb varijante aplikacije, kao što su identifikacija korisnika, više korisnika koji simultano učestvuju u istoj igri preko Interneta i pamćenje rezultata u bazi podataka na serveru. Na serveru se ne koristi baza podataka, već se svi resursi čuvaju u fajl sistemu.

Osim dopune aplikacije *Takmičenje* evidencijom rezultata na serveru i aplikacije *Editor* mogućnošću snimanja resursa na serveru, u novijim verzijama softvera se planira razvoj veb portala *Olimpijade znanja* koji bi omogućio registraciju i naja-

vu korisnika različitih uloga i privilegija, administraciju korisnika i tabli, igru više igrača preko Interneta, praćenje rezultata od strane različitih uloga, kao i korišćenje baze podataka na serveru za evidenciju korisnika i rezultata, kao i za čuvanje i razmenu resursa.

7. LITERATURA

- [1] Kipper, G., Rampolla, J., *Augmented Reality: An Emerging Technologies Guide to AR, First Edition*, Elsevier, 2013
- [2] Locke, J., *Some Thoughts Concerning Education*, 1693, http://playpen.meraka.csir.co.za/~acdc/education/Dr_Anvind_Gupa/Learners_Library_7_March_2007/Resources/books/John%20Locke.pdf (poslednji pristup: decembar 2013.)
- [3] Schell, J., *What Games Are Good At*, Games for Change 2012, <http://www.youtube.com/watch?v=sKVOiJ69sVw> (poslednji pristup: decembar 2013.)
- [4] Rice, J.W., "Educational Gaming", *What School Leaders Need to Know About Digital Technologies and Social Media*, ed. McLeod, S., Lehmann, C., Jossey-Bass, 2012, pp. 93-100
- [5] Slavković, M., *Razvoj softverskog sistema Olimpijada znanja, završni rad osnovnih akademskih studija*, 2010, <http://rti.etf.bg.ac.rs/rti/ri5rg/diplomski/SlavkovicMiroslav/završni.rad.pdf> (poslednji pristup: decembar 2013.)
- [6] Slavković, M., Đurđević, Đ., Tartalja, I., "Razvoj visoko prilagodljive igre Olimpijada znanja", *infoM, časopis za informacione tehnologije i multimedijalne sisteme*, 39/2011, str. 48-54
- [7] Steam, <http://store.steampowered.com/> (poslednji pristup: decembar 2013.)
- [8] Schell, J., *Visions of the Gamepocalypse*, FORA.tv, 2010, http://fora.tv/2010/07/27/Jesse_Schell_Visions_of_the_Gamepocalypse (poslednji pristup: decembar 2013.)
- [9] Slavković, M., *Funkcionalna specifikacija paketa Olimpijada znanja*, 2010, <http://rti.etf.bg.ac.rs/rti/ri5rg/diplomski/SlavkovicMiroslav/specifikacija.pdf> (poslednji pristup: decembar 2013.)
- [10] Mitra, N., *Digital Games: Computers at Play*, Infobase Publishing, 2010
- [11] Becker, K., "Distinctions Between Games and Learning: A Review of Current Literature on Games in Education", *Gaming and Cognition: Theories and Practice from the Learning Sciences*, ed. Eck, R.V., IGI Global, 2010, pp. 22-54
- [12] Tobias, S., Fletcher, J. D., "What research has to say about designing computer games for learning", *Educational Technology*, September–October 2007, pp. 20-29.
- [13] Horton, W., *E-Learning by Design*, 2nd Edition, Pfeiffer, 2012
- [14] Yannuzzi, T.J., Behrenshausen, B.G., "Serious Games for Transformative Learning: A Communication Perspective on the Radical Binarisation of Everyday Life", *Interdisciplinary Models and Tools for Serious Games: Emerging Concepts and Future Directions*, ed. Eck, R.V., IGI Global, 2010, pp. 74-102
- [15] Hacker, M., Kiggins, J., "Gaming to Learn – A promising Approach Using Educational Games to Stimulate STEM Learning", *Fostering Human Development Through Engineering and Technology Education*, ed. Barak, M., Hacker, M., Sense Publishers, 2011, pp. 257-276
- [16] Kapp, M.K., "A cultural perspective of teaching and learning ETE in digitally connected world", *Fostering Human Development Through Engineering and Technology Education*, ed. Barak, M., Hacker, M., Sense Publishers, 2011, pp. 207-232
- [17] Borchert, O., Brandt, L., Hokanson, G., Slator, B.M., Vender, B., Gutierrez, E.J., "Principles and Signatures in Serious Games for Science Education", *Gaming and Cognition: Theories and Practice from the Learning Sciences*, ed. Eck, R.V., IGI Global, 2010, pp. 312-338
- [18] Barrett, K.A., Johnson, W.L., "Developing Serious Games for Learning Language-in-Culture", *Gaming and Cognition: Theories and Practice from the Learning Sciences*, ed. Eck, R.V., IGI Global, 2010, pp. 281-311
- [19] Anderson, B., "MMORPGs in Support of Learning: Current Trends and Future Uses", *Gaming and Cognition: Theories and Practice from the Learning Sciences*, ed. Eck, R.V., IGI Global, 2010, pp. 55-80
- [20] Aremu, A., "Using 'TRIRACE©' in the Classroom: Perception on Modes and Effectiveness", *Gaming for Classroom-Based Learning: Digital Role Playing as a Motivator of Study*, ed. Baek, Y., IGI Global, 2010, pp. 66-83
- [21] Sauvé, L., "Using an Educational Online Game to Stimulate Learning", *Gaming for Classroom-Based Learning: Digital Role Playing as a Motivator of Study*, ed. Baek, Y., IGI Global, 2010, pp. 138-158
- [22] The try-with-resources Statement, <http://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html> (poslednji pristup: decembar 2013.)
- [23] Type Inference, <http://docs.oracle.com/javase/tutorial/java/generics/genTypeInference.html> (poslednji pristup: decembar 2013.)
- [24] Catching Multiple Exception Types and Rethrowing Exceptions with Improved Type Checking, <http://docs.oracle.com/javase/7/docs/technotes/guides/language/catch-multiple.html> (poslednji pristup: decembar 2013.)
- [25] Open Icon Library, <http://openiconlibrary.sourceforge.net/> (poslednji pristup: decembar 2013.)
- [26] One Laptop Per Child Free Sound Samples, http://wiki.laptop.org/go/Free_sound_samples (poslednji pristup: decembar 2013.)



Miroslav Slavković – Margintech d.o.o. Beograd U toku izrade rada, master student, Univerzitet u Beogradu - Elektrotehnički fakultet
Kontakt: apophys@sezampro.rs
Oblasti interesovanja: Veb dizajn i programiranje, Računarske igre



Dr Đorđe Đurđević, Univerzitet u Beogradu - Elektrotehnički fakultet
Kontakt: djordje.djurdjevic@etf.bg.ac.rs
Oblasti interesovanja: Računarska grafika, Objektno orijentisana analiza i dizajn, Digitalna obrada slike, e-Obrazovanje.



Dr Igor Tartalja, Univerzitet u Beogradu - Elektrotehnički fakultet
Kontakt: tartalja@etf.bg.ac.rs
Oblasti interesovanja: Objektno orijentisana analiza i projektovanje, Programski jezici i programiranje, Računarska grafika i vizuelizacija, Obrazovanje kroz zabavu, Automatizacija provere znanja